2011_normalized_metrics

Owen Liu

September 22, 2016

Following our meeting on September 20, 2016, we agreed on a reduced set of variables of interest, and agreed how to normalize them (to a 0 to 1 scale). This script performs that normalization and produces a clean dataset for mapping.

The variables are (S indicates it will be scaled/normalized, T means transformed, e.g. log-transformed):

- **TRADE DATA**
  - (S,T) Q.balance: Export/import ratio in quantity
  - (S,T) UV.balance: Export/import ratio in value, *per unit production*
  - (S,T) prod_ratio: Aquaculture/fisheries production (have to add this)
  - (S) spp_farmed: Total number of species harvested
- **FOOD SECURITY DATA**
  - (S,T) energy_adequacy: index of adequacy of the food supply in terms of calories
  - (S) gdp: GDP per capita
- **NUTRITION DATA**: all are percent from seafood/total nutrient intake
  - (S) polyunsatFA: polyunsatured fatty acids
  - (S) calories
  - (S) protein
  - (S) vitaminA
  - (S) thiamin
  - (S) niacin
  - (S) riboflavin
  - (S) B6
  - (S) iron
  - (S) calcium
  - (S) zinc
  - (S) vitamins_all: the mean value of scaled scores for vitamin A through zinc above (exluding FA, calories, and protein)
- **ECOLOGICAL DATA** (all need to be updated eventually)
  - (S) native: native vs. introduced species
  - (S) fishmeal: use of fishmeal in the diet
  - (S) trophic_level: species trophic level(*data does not exist yet*)
  - (S) food_conv: protein conversion ratio (*data does not exist yet*)
  - (S) habitat: local habitat/environmental impacts

Selecting and Joining Relevant Variables

Import the data

Select relevant variables

```
dat <- dat %>% select(country, year, Q.balance, UV.balance, energy_adequacy,
    gdp, polyunsatFA_percentseafood, calories_percentseafood,
    protein_percentseafood, vitaminA_percentseafood, thiamin_percentseafood,
    niacin_percentseafood, riboflavin_percentseafood, B6_percentseafood,
    iron_percentseafood, calcium_percentseafood, zinc_percentseafood,
    native, fishmeal, habitat)
```

Production ratio data

```
PD <- read.csv(paste0(W_D, "/data/Production datasets/PD.csv"))
```

Fix names to common:

```
# have to make sure country names line up
PD_names <- select(PD, country) %>% rename(PDname = country) %>%
    arrange(PDname) %>% distinct()
# write.csv(PD_names,file=paste0(W_D,'/data/nameconversion/PD_names.csv'),row.names=F)
names_conv <- read.csv(paste0(W_D, "/data/nameconversion/name_conversion.csv"),
    stringsAsFactors = F)
```

Join prod_ratio data

```
dat2 <- dat %>% left_join(PD, by = c(year = "year", country = "country")) %>%
    select(-(X:a.production)) %>% rename(prod_ratio = p.ratio,
    spp_farmed = a.species)

# New dataset for normalized data
dat.norm <- dat2
```

Variable Normalization

Trade metrics

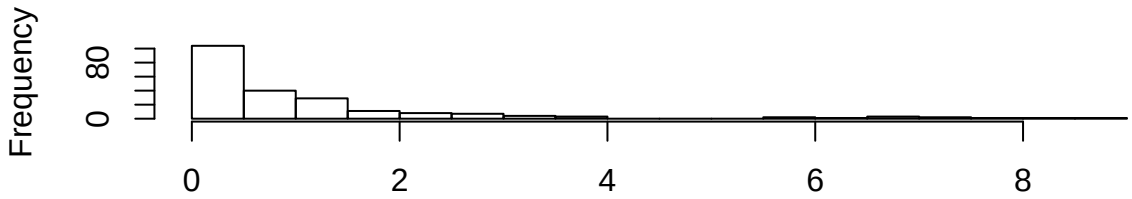
Normalize trade metrics. Normalization for for Q.balance and UV.balance is

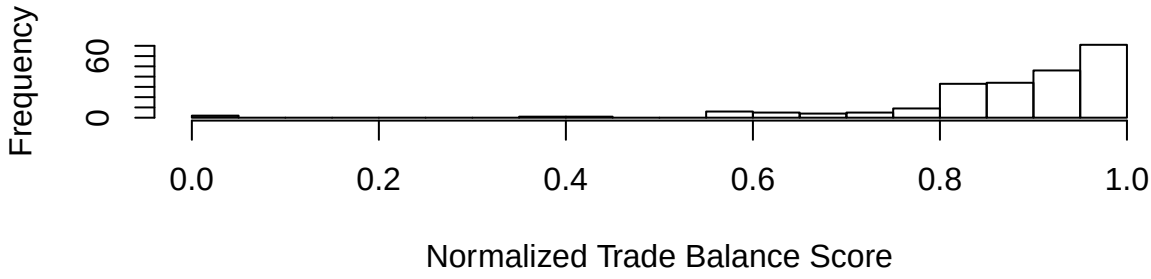
$$Norm = |\frac{|\log Raw|}{\max|\log Raw|} - 1|$$

In this formulation, the absolute value of the log means that a one to one ratio will be 0, and the distance from that value (in either direction) will be greater than 0. We then divide by the maximum and take the absolute value of (that value minus 1) to switch the scale from 1 to 0 to 0 to 1. Taking the log, while somewhat distorting the data, also reduces the impact of outliers. In the end, a score of 0 is those values furthest from a perfect ratio, and a score of 1 is a perfect 1:1 trade balance.

```
norm_trade <- function(x) {
    nrm <- abs(log10(x))
    nrm[is.infinite(nrm)] <- NA
    out <- abs(nrm/max(nrm, na.rm = T) - 1)
    return(out)
}
# compare distribution of raw/normalized data, for Q.balance
test <- norm_trade(dat2$Q.balance)
```

```
par(mfrow = c(2, 1))
hist(dat2$Q.balance, xlab = "Raw Trade Balance (Exports/Imports in tonnes)",
    main = "", breaks = 20)
hist(test, xlab = "Normalized Trade Balance Score", main = "",
    breaks = 20)
```



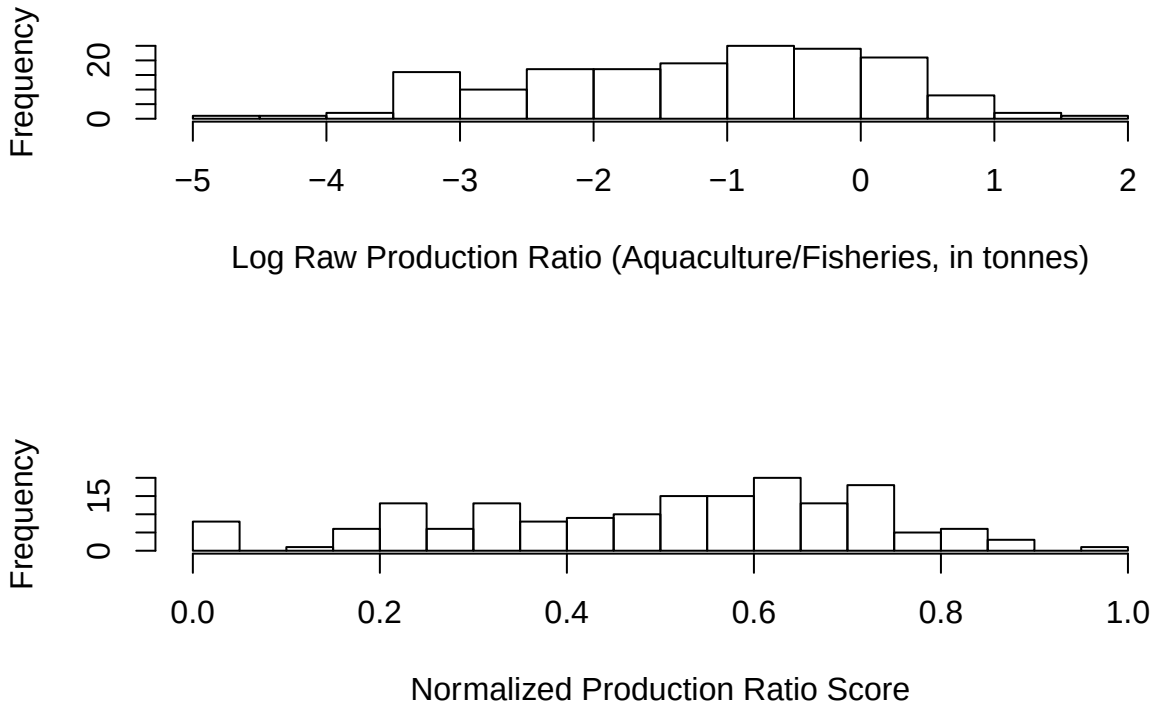



```
# Looks good
dat.norm <- mutate(dat.norm, Q.balance = norm_trade(Q.balance),
    UV.balance = norm_trade(UV.balance))
```

Production balance, prod_ratio, will be scaled to where 1 and 0 are equal to the maximum and minimum observed proportion of aquaculture to fisheries, respectively. We also take a log here to reduce outliers. Negative infinite values for the log (corresponding to 0 aquaculture production) receive a score of 0.

```
norm_prod_ratio <- function(x) {
    nrm <- log10(x)
    out <- (nrm - min(nrm[is.finite(nrm)]))/(max(nrm[is.finite(nrm)]) -
        min(nrm[is.finite(nrm)]))
    out[is.infinite(nrm)] <- 0
    return(out)
}
# compare distribution of raw/normalized data, for prod_ratio
```

```
test <- norm_prod_ratio(dat2$prod_ratio)
par(mfrow = c(2, 1))
hist(log10(dat2$prod_ratio), xlab = "Log Raw Production Ratio (Aquaculture/Fisheries, in tonnes)",
    main = "", breaks = 20)
hist(test, xlab = "Normalized Production Ratio Score", main = "",
    breaks = 20)
```

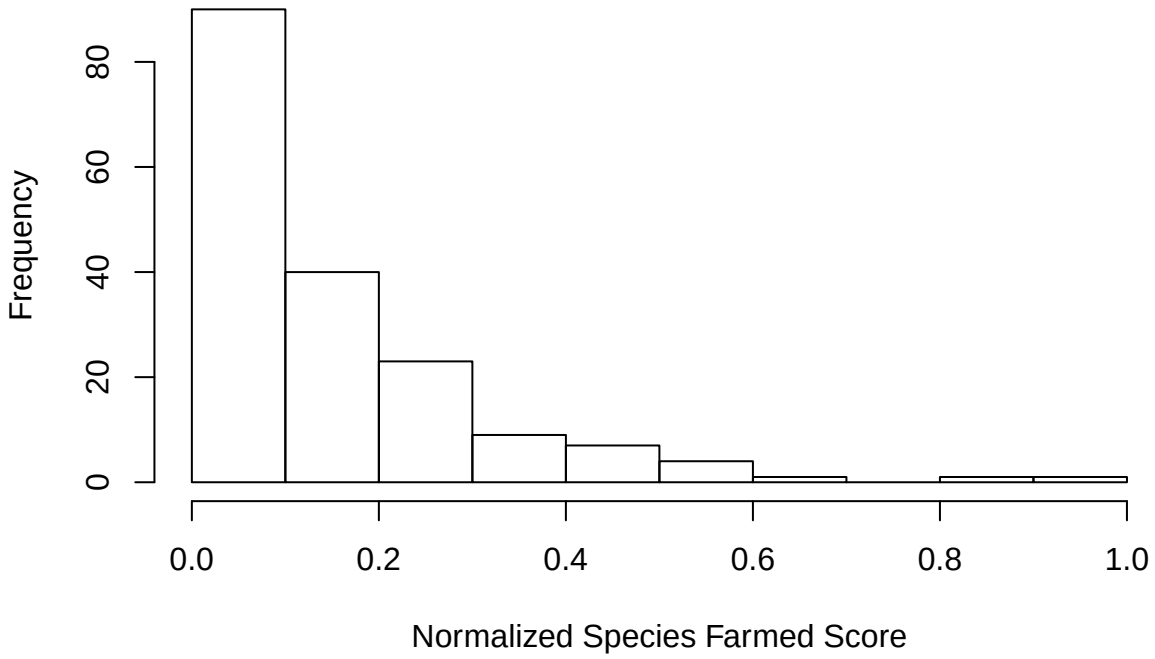



```
dat.norm <- mutate(dat.norm, prod_ratio = norm_prod_ratio(prod_ratio))
```

Not sure how much I like this one (normalization on top of log transformation really distorts the scale to where a country with still a very small production ratio).

Total number of species harvested, relative to the most observed.

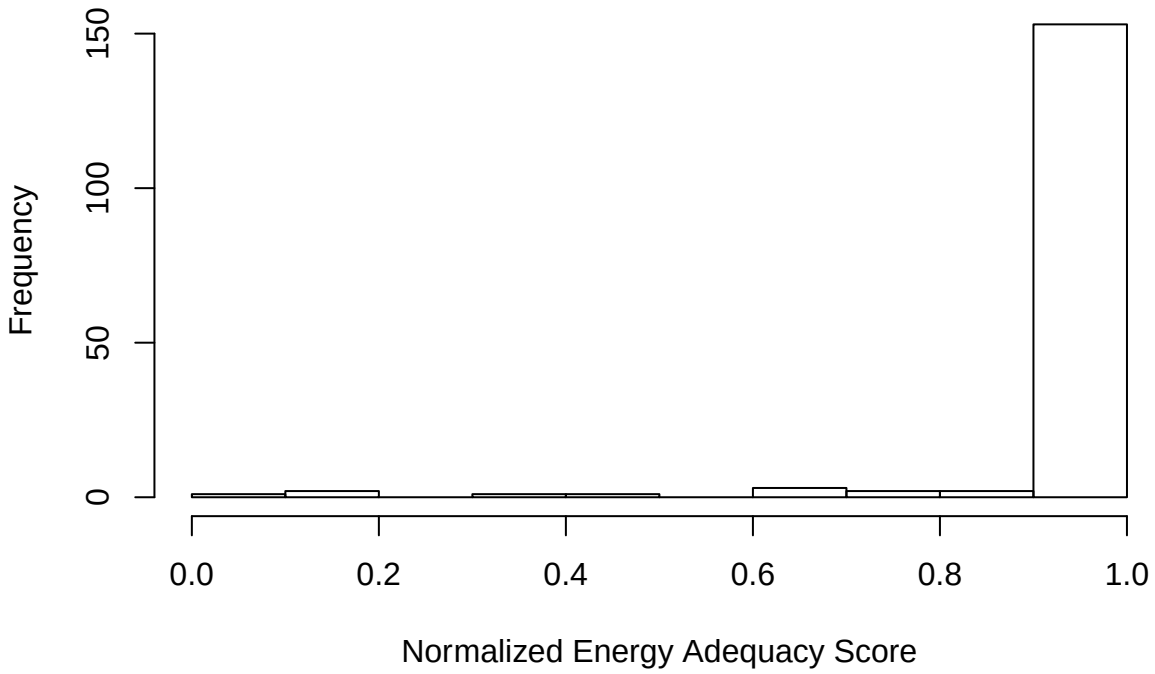
```
normalize <- function(x) (x - min(x, na.rm = T))/(max(x, na.rm = T) -
    min(x, na.rm = T))
dat.norm <- mutate(dat.norm, spp_farmed = normalize(spp_farmed))

par(mfrow = c(1, 1))
hist(dat.norm$spp_farmed, xlab = "Normalized Species Farmed Score",
    main = "")
```



Food Security Data

energy_adequacy, scaled such that 100% or over is equal to 1.

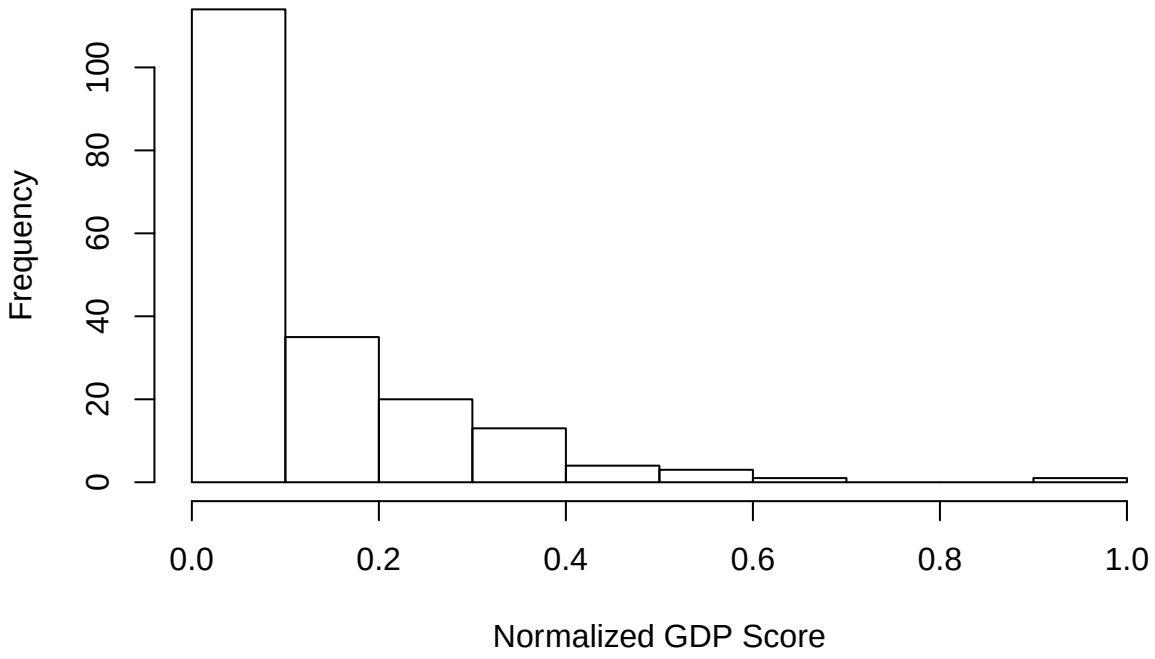
```
norm_energy_ad <- function(x) {
    x[x > 100] <- 100
    out <- normalize(x)
    return(out)
}
dat.norm <- mutate(dat.norm, energy_adequacy = norm_energy_ad(energy_adequacy))
hist(dat.norm$energy_adequacy, xlab = "Normalized Energy Adequacy Score",
    main = "")
```



The problem here is that most diets are energy adequate...

GDP per capita, scaled relative to richest country.

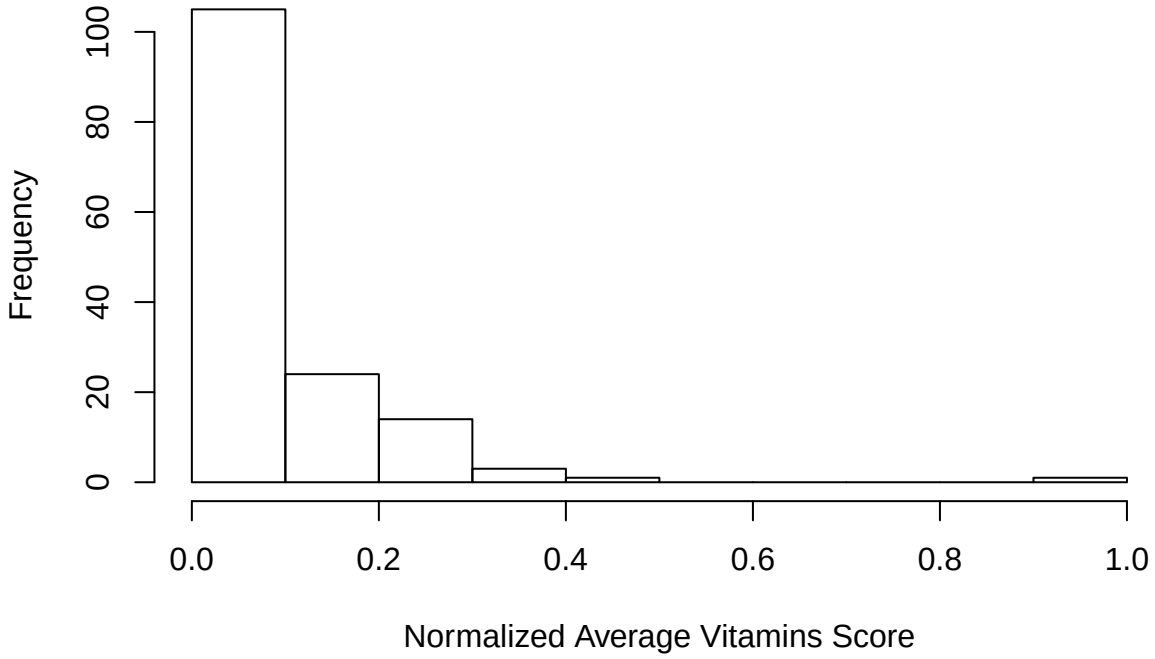
```
dat.norm <- mutate(dat.norm, gdp = normalize(gdp))
hist(dat.norm$gdp, xlab = "Normalized GDP Score", main = "")
```



Nutrition Data

We first need to combine all of the vitamins into one metric. We do this by normalizing them individually and then taking a mean for each country across the individual scores

```
# Scale each vitamin
dat.norm <- dat.norm %>% mutate_each(funs(normalize), vitaminA_percentseafood:zinc_percentseafood)

# Calculate a mean vitamin score
dat.norm$vitamins_all <- dat.norm %>% select(vitaminA_percentseafood:zinc_percentseafood) %>%
    rowMeans(na.rm = TRUE)

dat.norm$vitamins_all[is.nan(dat.norm$vitamins_all)] <- NA

hist(dat.norm$vitamins_all, main = "", xlab = "Normalized Average Vitamins Score")
```



Protein, calories, fatty acids all scaled relative to max.

```
dat.norm <- dat.norm %>% mutate_each(funs(normalize), polyunsatFA_percentseafood:protein_percentseafood
```

Ecological Data

All of the ecological metrics are scaled to 0 to 10. We just redefine this to a 0 to 1 scale.

```
dat.norm <- dat.norm %>% mutate_each(funs(normalize), native:habitat)
```

Output data

Write output

```
write.csv(dat.norm, file = paste0(W_D, "/data/data_normalized_092616.csv"))
```